CS 473: Algorithms, Fall 2008
HW 5 (due Tuesday, October 14, 11am)

This homework contains four problems. **Read the instruction for submitting homework on the course webpage**. In particular, *make sure* that you write the solutions for the problems on separate sheets of paper and then staple them together. Write your name and netid on each sheet.

Collaboration Policy: For this homework you are allowed to work in groups of up to 3 students each. Starting this week, a third of the on-campus students will be presenting their homework orally. Please see the newgroup for instructions on which groups will be presenting orally and the instructions for signing up for a slot. The other groups will submit a written homework.

1. (25 pts) You are given an undirected graph $G = (V, E)$ with non-negative edge weights: $w(e)$ denotes the weight of edge e. Let m and n be the number of edges and nodes in G respectively. Suppose you are told that the graph has at most k *distinct* edge weights.

   - Describe an algorithm that first checks whether there are only k distinct edge weights or not in $O(m \log k)$ time.
   - Describe an algorithm that finds the MST of G in $O((m + n) \log k)$ time. *Hint:* Use Prim's algorithm with appropriate data structure modifications.

   Partial credit for an algorithm that runs in $O((m + n) \cdot k)$ time for both of the above steps.

2. (25 pts) You and your friend exchange communicate using an alphabet Σ with $n = \Sigma$. Both of you have a previously agreed upon fixed length encoding of Σ in which each letter a is encoded by $\lceil \log n \rceil$ bits. For example Σ could be all the ASCII letters and both you and your friend know ASCII. You now want to send a big file to your friend and decide to use prefix coding. Having just learnt that Huffman encoding gives optimal prefix codes, you first evaluated the frequencies/probabilities of the letters in the file (say p_i is the probability of the i'th letter) and used Huffman's algorithm to compute the optimal code. Now you need to send this code to your friend in as short a message as possible so that he/she can decode once the file is transmitted (your friend does not know the probabilities of the letters). Two simple ideas for sending the code are as follows. First idea is to to send for each $a \in \Sigma$ the string $\gamma(a)$ that is computed by the algorithm; with a separator between each string. This may take $\Omega(n^2)$ bits (do you see an example?). The second option is to send the $p_1, p_2, \ldots, p_n$ to your friend so that he/she can run the Huffman coding algorithm to compute the code. This can also take $\Omega(n^2)$ bits (in fact much more if some probabilities are really tiny). Show that you can send a message of $O(n \log n)$ bits to your friend such that he/she can find out your Huffman code.

3. (25 pts) Recall from class the idea of Path-Compression in the Union-Find data structure for maintaining disjoint sets. Prove that if Path-Compression is used then the total time for k operations in which all the `union` operations precede all the `find` operations, is $O(k)$. (The cost of initializing the data structure via `makeUnionFind` is not part of the cost of the k operations). Note that the running time is independent of the number of elements n.

4. (25 pts) Exercise 6.1 from the text book.